Project Design Phase-II
Technology Stack (Architecture & Stack)

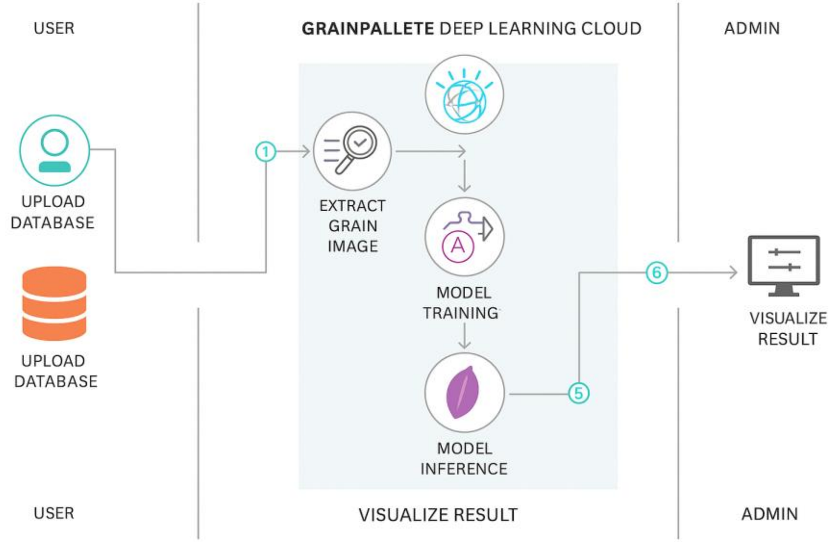
Date	31 January 3035
Team ID	LTVIP2025TMID33633
Project Name	GrainPalette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



Guidelines:

1. **Application Logic / Technology Blocks**
   - User uploads rice image
   - Image is processed
   - Model predicts rice type
   - Result is displayed with confidence score
2. **Infrastructural Demarcation**
   - Model training: Local or Google Colab
   - Model can also run offline with .h5 file
3. **External Interfaces**
   - Google Drive API (for dataset storage – optional)
   - GitHub (for code repository)
   - No third-party APIs used in prediction
4. **Data Storage Components**
   - Images stored locally or in cloud drive
   - Trained model stored as .h5 file
   - Prediction logs can be stored in CSV or Firebase
   - Model predicts and returns rice type and confidence

Table-1: Components & Technologies – GrainPalette

S.No	Component	Description	Technology
1.	User Interface	Web interface for user to upload rice image & view result	HTML, CSS, Streamlit (or Flask + HTML)
2.	Application Logic-1	Image preprocessing and UI logic	Python, Flask
3.	Application Logic-2	Image classification logic using deep learning	TensorFlow, Keras
4.	Application Logic-3	Model loading and confidence scoring	Keras Model Inference
5.	Database	Local prediction history (optional)	SQLite / CSV
6.	Cloud Database	(Optional) for storing predictions in the cloud	Firebase / Firestore
7.	File Storage	Storing model and input/output images	Local Filesystem
8.	External API-1	(Optional) For cloud storage access	Google Drive API (Optional)
9.	External API-2	(Optional) Email/SMS alerts or notifications	Not used currently
10.	Machine Learning Model	Predicts rice grain type from image	MobileNetV4 (Transfer Learning)
11.	Infrastructure (Server/Cloud)	Deployed and tested on local and cloud platform	Local System, Streamlit Cloud / FlaskApp

Table-2: Application Characteristics – GrainPalette

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Used open-source tools and libraries for development	Flask, TensorFlow, Keras, Streamlit
2.	Security Implementations	No user login; local use only. Can be extended using JWT, API keys for APIs	(Planned) SHA-256, Flask-Security
3.	Scalable Architecture	Modular architecture allows adding more rice types or new crop models	3-tier (UI – Logic – Model)
4.	Availability	Cloud hosting ensures app can run anytime; model can be used offline too	Streamlit Cloud, Local System
5.	Performance	Pre-trained MobileNet model ensures fast and accurate predictions	Keras, TensorFlow

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d